

# Face Recognition Using Laplacianfaces

Xiaofei He<sup>†</sup>, Shuicheng Yan<sup>‡</sup>, Yuxiao Hu<sup>\*</sup>, Partha Niyogi<sup>†</sup>, Hong-Jiang Zhang<sup>\*</sup>

<sup>†</sup>Department of Computer Science, The University of Chicago, 1100 E. 58<sup>th</sup> Street, Chicago, IL 60637  
{xiaofei, niyogi}@cs.uchicago.edu

<sup>\*</sup>Microsoft Research Asia, Beijing 100080, China  
{i-yuxhu, hjzhang}@microsoft.com

<sup>‡</sup>Department of Information Science, School of Mathematical Sciences, Beijing University, China  
scyan@msrchina.research.microsoft.com

## ABSTRACT

*We propose an appearance based face recognition method called the Laplacianface approach. By using Locality Preserving Projections (LPP), the face images are mapped into a face subspace for analysis. Different from Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) which effectively see only the Euclidean structure of face space, LPP finds an embedding that preserves local information, and obtains a face subspace that best detects the essential face manifold structure. The Laplacianfaces are the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the face manifold. In this way, the unwanted variations resulting from changes in lighting, facial expression, and pose may be eliminated or reduced. Theoretical analysis shows that PCA, LDA and LPP can be obtained from different graph models. We compare the proposed Laplacianface approach with Eigenface and Fisherface methods on three different face datasets. Experimental results suggest that the proposed Laplacianface approach provides a better representation and achieves lower error rates in face recognition.*

## KEYWORDS

Face Recognition, Principal Component Analysis, Linear Discriminant Analysis, Locality Preserving Projections, Face Manifold, Subspace Learning

# 1. INTRODUCTION

Many face recognition techniques have been developed over the past few decades. One of the most successful and well-studied techniques to face recognition is the *appearance-based* method [28][16]. When using appearance-based methods, we usually represent an image of size  $n \times m$  pixels by a vector in an  $n \times m$  dimensional space. In practice, however, these  $n \times m$  dimensional spaces are too large to allow robust and fast face recognition. A common way to attempt to resolve this problem is to use dimensionality reduction techniques [1][2][8][11][12][14][22][26][28][32][35]. Two of the most popular techniques for this purpose are Principal Component Analysis (PCA) [28] and Linear Discriminant Analysis (LDA) [2].

PCA is an eigenvector method designed to model linear variation in high-dimensional data. PCA performs dimensionality reduction by projecting the original  $n$ -dimensional data onto the  $k$  ( $\ll n$ )-dimensional linear subspace spanned by the leading eigenvectors of the data's covariance matrix. Its goal is to find a set of mutually orthogonal basis functions that capture the directions of maximum variance in the data and for which the coefficients are pairwise decorrelated. For linearly embedded manifolds, PCA is guaranteed to discover the dimensionality of the manifold and produces a compact representation. Turk and Pentland [28] use Principal Component Analysis to describe face images in terms of a set of basis functions, or "eigenfaces".

LDA is a supervised learning algorithm. LDA searches for the project axes on which the data points of different classes are far from each other while requiring data points of the same class to be close to each other. Unlike PCA which encodes information in an orthogonal linear space, LDA encodes discriminating information in a linear separable space using bases are not necessarily orthogonal. It is generally believed that algorithms based on LDA are superior to those based on PCA. However, some recent work [14] shows that, when the training dataset is small, PCA can outperform LDA, and also that PCA is less sensitive to different training datasets.

Recently, a number of research efforts have shown that the face images possibly reside on a nonlinear submanifold [7][10][18][19][21][23][27]. However, both PCA and LDA effectively see only the Euclidean structure. They fail to discover the underlying structure, if the face images lie on a nonlinear submanifold hidden in the image space. Some nonlinear techniques have been proposed to discover the nonlinear structure of the manifold, *e.g.* Isomap [27], LLE [18][20], and Laplacian Eigenmap [3]. These nonlinear methods do yield impressive results on some benchmark artificial data sets. However, they yield maps that are defined *only* on the training data points and how to evaluate the maps on novel test data points remains unclear. Therefore, these nonlinear manifold learning techniques [3][5][18][20][27][33] might not be suitable for some computer vision tasks, such as face recognition.

In the meantime, there has been some interest in the problem of developing low dimensional representations through kernel based techniques for face recognition [13][19]. These methods can discover the nonlinear structure of the face images. However, they are computationally expensive. Moreover, none of them explicitly considers the structure of the manifold on which the face images possibly reside.

In this paper, we propose a new approach to face analysis (representation and recognition), which explicitly considers the manifold structure. To be specific, the manifold structure is modeled by a nearest-neighbor graph which preserves the local structure of the image space. A face subspace is obtained by *Locality Preserving Projections* (LPP) [9]. Each face image in the image space is mapped to a low-dimensional face subspace, which is characterized by a set of feature images, called *Laplacianfaces*. The face subspace preserves local structure, seems to have more discriminating power than the PCA approach for classification purpose. We also provide theoretical analysis to show that PCA, LDA and LPP can be obtained from different graph models. Central to this is a graph structure that is induced from the data points. LPP finds a projection that respects this graph structure. In our theoretical analysis, we show how PCA, LDA, and LPP arise from the same principle applied to different choices of this graph structure.

It is worthwhile to highlight several aspects of the proposed approach here:

1. While the Eigenfaces method aims to preserve the global structure of the image space, and the Fisherfaces method aims to preserve the discriminating information; our Laplacianfaces method aims to preserve the local structure of the image space. In many real world classification problems, the local manifold structure is more important than the global Euclidean structure, especially when nearest-neighbor like classifiers are used for classification. LPP seems to have discriminating power although it is unsupervised.
2. An efficient subspace learning algorithm for face recognition should be able to discover the nonlinear manifold structure of the face space. Our proposed Laplacianfaces method explicitly considers the manifold structure which is modeled by an adjacency graph. Moreover, the *Laplacianfaces* are obtained by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the face manifold. They reflect the intrinsic face manifold structures.
3. LPP shares some similar properties to LLE, such as a locality preserving character. However, their objective functions are totally different. LPP is obtained by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the manifold. LPP is linear, while LLE is nonlinear. Moreover, LPP is defined everywhere, while LLE is defined only on the training data points and it is unclear how to evaluate the maps for new test points. In contrast, LPP may be simply applied to any new data point to locate it in the reduced representation space.

The rest of this paper is organized as follows: Section 2 describes PCA and LDA. The *Locality Preserving Projections* (LPP) algorithm is described in Section 3. In Section 4, we provide a statistical view of LPP. We then give a theoretical analysis of LPP and its connections to PCA and LDA in Section 5. Section 6 presents the manifold ways of face analysis using *Laplacianfaces*. A variety of experimental results are presented in Section 7. Finally, we provide some concluding remarks and suggestions for future work in Section 8.

## 2. PCA and LDA

One approach to coping with the problem of excessive dimensionality of the image space is to reduce the dimensionality by combining features. Linear combinations are particularly attractive because they are simple to compute and analytically tractable. In effect, linear methods project the high-dimensional data onto a lower dimensional subspace.

Considering the problem of representing all of the vectors in a set of  $n$   $d$ -dimensional samples  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ , with zero mean, by a single vector  $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$  such that  $y_i$  represent  $\mathbf{x}_i$ . Specifically, we find a linear mapping from the  $d$ -dimensional space to a line. Without loss of generality, we denote the transformation vector by  $\mathbf{w}$ . That is,  $\mathbf{w}^T \mathbf{x}_i = y_i$ . Actually, the magnitude of  $\mathbf{w}$  is of no real significance, because it merely scales  $y_i$ . In face recognition, each vector  $\mathbf{x}_i$  denotes a face image.

Different objective functions will yield different algorithms with different properties. PCA aims to extract a subspace in which the variance is maximized. Its objective function is as follows,

$$\max_{\mathbf{w}} \sum_{i=1}^n (y_i - \bar{y})^2$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

The output set of principal vectors  $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k$  are an orthonormal set of vectors representing the eigenvectors of the sample covariance matrix associated with the  $k < d$  largest eigenvalues.

While PCA seeks directions that are efficient for representation, Linear Discriminant Analysis seeks directions that are efficient for discrimination. Suppose we have a set of  $n$   $d$ -dimensional samples  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ , belonging to  $l$  classes of faces. The objective function is as follows,

$$\max_{\mathbf{w}} \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$$

$$S_B = \sum_{i=1}^l n_i (\mathbf{m}^{(i)} - \mathbf{m})(\mathbf{m}^{(i)} - \mathbf{m})^T$$

$$S_W = \sum_{i=1}^l \left( \sum_{j=1}^{n_i} (\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})(\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})^T \right)$$

where  $\mathbf{m}$  is the total sample mean vector,  $n_i$  is the number of samples in the  $i^{\text{th}}$  class,  $\mathbf{m}^{(i)}$  is the average vector of the  $i^{\text{th}}$  class, and  $\mathbf{x}_j^{(i)}$  is the  $j^{\text{th}}$  sample in the  $i^{\text{th}}$  class. We call  $S_W$  the *within-class scatter matrix* and  $S_B$  the *between-class scatter matrix*.

### 3. LEARNING A LOCALITY PRESERVING SUBSPACE

PCA and LDA aim to preserve the global structure. However, in many real world applications, the local structure is more important. In this section, we describe *Locality Preserving Projection* (LPP) [9], a new algorithm for learning a locality preserving subspace. The complete derivation and theoretical justifications of LPP can be traced back to [9]. LPP seeks to preserve the intrinsic geometry of the data and local structure. The objective function of LPP is as follows:

$$\min \sum_{ij} (y_i - y_j)^2 S_{ij}$$

where  $y_i$  is the one-dimensional representation of  $\mathbf{x}_i$  and the matrix  $S$  is a similarity matrix. A possible way of defining  $S$  is as follows:

$$S_{ij} = \begin{cases} \exp(\|\mathbf{x}_i - \mathbf{x}_j\|^2 / t), & \|\mathbf{x}_i - \mathbf{x}_j\|^2 < \varepsilon \\ 0 & \text{otherwise.} \end{cases}$$

or

$$S_{ij} = \begin{cases} \exp(\|\mathbf{x}_i - \mathbf{x}_j\|^2 / t), & \text{if } \mathbf{x}_i \text{ is among } k \text{ nearest neighbors of } \mathbf{x}_j \\ & \text{or } \mathbf{x}_j \text{ is among } k \text{ nearest neighbors of } \mathbf{x}_i \\ 0 & \text{otherwise.} \end{cases}$$

where  $\varepsilon$  is sufficiently small, and  $\varepsilon > 0$ . Here,  $\varepsilon$  defines the radius of the local neighborhood. In other words,  $\varepsilon$  defines the “locality”. The objective function with our choice of symmetric weights  $S_{ij}$  ( $S_{ij} = S_{ji}$ ) incurs a heavy penalty if neighboring points  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are mapped far apart, *i.e.* if  $(y_i - y_j)^2$  is large. Therefore, minimizing it is an attempt to ensure that if  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are “close” then  $y_i$  and  $y_j$  are close as well. Following some simple algebraic steps, we see that

$$\begin{aligned}
& \frac{1}{2} \sum_{ij} (y_i - y_j)^2 S_{ij} \\
&= \frac{1}{2} \sum_{ij} (\mathbf{w}^T \mathbf{x}_i - \mathbf{w}^T \mathbf{x}_j)^2 S_{ij} \\
&= \sum_{ij} \mathbf{w}^T \mathbf{x}_i S_{ij} \mathbf{x}_i^T \mathbf{w} - \sum_{ij} \mathbf{w}^T \mathbf{x}_i S_{ij} \mathbf{x}_j^T \mathbf{w} \\
&= \sum_i \mathbf{w}^T \mathbf{x}_i D_{ii} \mathbf{x}_i^T \mathbf{w} - \mathbf{w}^T X S X^T \mathbf{w} \\
&= \mathbf{w}^T X D X^T \mathbf{w} - \mathbf{w}^T X S X^T \mathbf{w} \\
&= \mathbf{w}^T X (D - S) X^T \mathbf{w} \\
&= \mathbf{w}^T X L X^T \mathbf{w}
\end{aligned}$$

where  $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ , and  $D$  is a diagonal matrix; its entries are column (or row, since  $S$  is symmetric) sums of  $S$ ,  $D_{ii} = \sum_j S_{ji}$ .  $L = D - S$  is the Laplacian matrix [6]. Matrix  $D$  provides a natural measure on the data points. The bigger the value  $D_{ii}$  (corresponding to  $y_i$ ) is, the more “important” is  $y_i$ . Therefore, we impose a constraint as follows:

$$\begin{aligned}
& \mathbf{y}^T D \mathbf{y} = 1 \\
\Rightarrow & \mathbf{w}^T X D X^T \mathbf{w} = 1
\end{aligned}$$

Finally, the minimization problem reduces to finding:

$$\begin{aligned}
& \arg \min_{\mathbf{w}} \mathbf{w}^T X L X^T \mathbf{w} \\
& \mathbf{w}^T X D X^T \mathbf{w} = 1
\end{aligned}$$

The transformation vector  $\mathbf{w}$  that minimizes the objective function is given by the minimum eigenvalue solution to the generalized eigenvalue problem:

$$X L X^T \mathbf{w} = \lambda X D X^T \mathbf{w}$$

Note that the two matrices  $XLX^T$  and  $XD\mathbf{X}^T$  are both symmetric and positive semi-definite, since the Laplacian matrix  $L$  and the diagonal matrix  $D$  are both symmetric and positive semi-definite.

The Laplacian matrix for finite graph is analogous to the Laplace Beltrami operator on compact Riemannian manifolds. While the Laplace Beltrami operator for a manifold is generated by the Riemannian metric, for a graph it comes from the adjacency relation. Belkin and Niyogi [3] showed that the optimal map preserving locality can be found by solving the following optimization problem on the manifold:

$$\min_{\|f\|_{L^2(M)}=1} \int_M \|\nabla f\|^2$$

which is equivalent to

$$\min_{\|f\|_{L^2(M)}=1} \int_M \mathcal{L}(f)f$$

where  $\mathcal{L}$  is the Laplace Beltrami operator on the manifold, i.e.  $\mathcal{L}(f) = -\operatorname{div}\nabla(f)$ . Thus, the optimal  $f$  has to be an eigenfunction of  $\mathcal{L}$ . If we assume  $f$  to be linear, we have  $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$ . By spectral graph theory, the integral can be discretely approximated by  $\mathbf{w}^T XLX^T \mathbf{w}$  and the  $L^2$  norm of  $f$  can be discretely approximated by  $\mathbf{w}^T XD\mathbf{X}^T \mathbf{w}$ , which will ultimately lead to the following eigenvalue problem:

$$XLX^T \mathbf{w} = \lambda XD\mathbf{X}^T \mathbf{w}$$

The derivation reflects the intrinsic geometric structure of the manifold. For details, see [3][6][9].

#### 4. STATISTICAL VIEW OF LPP

LPP can also be obtained from statistical viewpoint. Suppose the data points follow some underlying distribution. Let  $\mathbf{x}$  and  $\mathbf{y}$  be two random variables. We define that a linear mapping  $\mathbf{x} \rightarrow \mathbf{w}^T \mathbf{x}$  best preserves the local structure of the underlying distribution in the  $L^2$  sense if it minimizes the  $L^2$  distances between  $\mathbf{w}^T \mathbf{x}$  and  $\mathbf{w}^T \mathbf{y}$  provided that  $\|\mathbf{x} - \mathbf{y}\| < \varepsilon$ . Namely:

$$\min E(|\mathbf{w}^T \mathbf{x} - \mathbf{w}^T \mathbf{y}|^2 \mid \|\mathbf{x} - \mathbf{y}\| < \varepsilon)$$

where  $\varepsilon$  is sufficiently small, and  $\varepsilon > 0$ . Here,  $\varepsilon$  defines the “locality”. Define  $\mathbf{z} = \mathbf{x} - \mathbf{y}$ , then we have the following objective function:

$$\min E(|\mathbf{w}^T \mathbf{z}|^2 \mid \|\mathbf{z}\| < \varepsilon)$$

It follows that,

$$\begin{aligned} & E(|\mathbf{w}^T \mathbf{z}|^2 \mid \|\mathbf{z}\| < \varepsilon) \\ &= E(\mathbf{w}^T \mathbf{z} \mathbf{z}^T \mathbf{w} \mid \|\mathbf{z}\| < \varepsilon) \\ &= \mathbf{w}^T E(\mathbf{z} \mathbf{z}^T \mid \|\mathbf{z}\| < \varepsilon) \mathbf{w} \end{aligned}$$

Given a set of sample points  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ , we first define an indicator function  $S_{ij}$  as follows:

$$S_{ij} = \begin{cases} 1, & \|\mathbf{x}_i - \mathbf{x}_j\|^2 < \varepsilon \\ 0 & \text{otherwise.} \end{cases}$$

Let  $d$  be the number of non-zero  $S_{ij}$ , and  $D$  be a diagonal matrix whose entries are column (or row, since  $S$  is symmetric) sums of  $S$ ,  $D_{ii} = \sum_j S_{ji}$ . By the Strong Law of Large Numbers,  $E(\mathbf{z} \mathbf{z}^T \mid \|\mathbf{z}\| < \varepsilon)$  can be estimated from the sample points as follows:

$$\begin{aligned} & E(\mathbf{z} \mathbf{z}^T \mid \|\mathbf{z}\| < \varepsilon) \\ &\approx \frac{1}{d} \sum_{\|\mathbf{z}\| < \varepsilon} \mathbf{z} \mathbf{z}^T \\ &= \frac{1}{d} \sum_{\|\mathbf{x}_i - \mathbf{x}_j\| < \varepsilon} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \\ &= \frac{1}{d} \sum_{i,j} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T S_{ij} \\ &= \frac{1}{d} \left( \sum_{i,j} \mathbf{x}_i \mathbf{x}_i^T S_{ij} + \sum_{i,j} \mathbf{x}_j \mathbf{x}_j^T S_{ij} - \sum_{i,j} \mathbf{x}_i \mathbf{x}_j^T S_{ij} - \sum_{i,j} \mathbf{x}_j \mathbf{x}_i^T S_{ij} \right) \\ &= \frac{2}{d} \left( \sum_i \mathbf{x}_i \mathbf{x}_i^T D_{ii} - \sum_{i,j} \mathbf{x}_i \mathbf{x}_j^T S_{ij} \right) \\ &= \frac{2}{d} (XDX^T - XSX^T) \\ &= \frac{2}{d} XLX^T \end{aligned}$$

where  $L = D - S$  is the Laplacian matrix. The  $i^{\text{th}}$  column of matrix  $X$  is  $\mathbf{x}_i$ . By imposing the same constraint, we finally get the same minimization problem described in Section 3.

## 5. THEORETICAL ANALYSIS OF LPP, PCA AND LDA

In this section, we present a theoretical analysis of LPP and its connections to PCA and LDA.

### 5.1 Connections to PCA

We first discuss the connections between LPP and PCA.

It is worthwhile to point out that  $XLX^T$  is the data covariance matrix, if the Laplacian matrix  $L$  is  $\frac{1}{n}I - \frac{1}{n^2}\mathbf{e}\mathbf{e}^T$ , where  $n$  is the number of data points,  $I$  is the identity matrix and  $\mathbf{e}$  is a column vector taking 1 at each entry. In fact, the Laplacian matrix here has the effect of removing the sample mean from the sample vectors. In this case, the weight matrix  $S$  takes  $1/n^2$  at each entry, i.e.  $S_{ij} = 1/n^2, \forall i, j$ .  $D_{ii} = \sum_j S_{ji} = 1/n$ . Hence the Laplacian matrix is  $L = D - S = \frac{1}{n}I - \frac{1}{n^2}\mathbf{e}\mathbf{e}^T$ . Let  $\mathbf{m}$  denote the sample mean, i.e.  $\mathbf{m} = 1/n\sum_i \mathbf{x}_i$ . We have:

$$\begin{aligned}
 XLX^T &= \frac{1}{n}X\left(I - \frac{1}{n}\mathbf{e}\mathbf{e}^T\right)X^T \\
 &= \frac{1}{n}XX^T - \frac{1}{n^2}(X\mathbf{e})(X\mathbf{e})^T \\
 &= \frac{1}{n}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{n^2}(n\mathbf{m})(n\mathbf{m})^T \\
 &= \frac{1}{n}\sum_i (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T + \frac{1}{n}\sum_i \mathbf{x}_i\mathbf{m}^T + \frac{1}{n}\sum_i \mathbf{m}\mathbf{x}_i^T - \frac{1}{n}\sum_i \mathbf{m}\mathbf{m}^T - \mathbf{m}\mathbf{m}^T \\
 &= E[(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T] + 2\mathbf{m}\mathbf{m}^T - 2\mathbf{m}\mathbf{m}^T \\
 &= E[(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T]
 \end{aligned}$$

where  $E[(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T]$  is just the covariance matrix of the data set.

The above analysis shows that the weight matrix  $S$  plays a key role in the LPP algorithm. When we aim at preserving the global structure, we take  $\varepsilon$  (or  $k$ ) to be infinity and choose the eigenvectors (of the matrix  $XLX^T$ ) associated with the largest eigenvalues. Hence the data points are projected along the directions of maximal variance. When we aim at preserving the local structure, we take  $\varepsilon$  to be sufficiently small and choose the eigenvectors (of the matrix  $XLX^T$ ) associated with the smallest eigenvalues. Hence the data points are projected along the directions preserving locality. It is important to note that,

when  $\varepsilon$  (or  $k$ ) is sufficiently small, the Laplacian matrix is no longer the data covariance matrix, and hence the directions preserving locality are not the directions of minimal variance. In fact, the directions preserving locality are those minimizing *local* variance.

## 5.2 Connections to LDA

LDA seeks directions that are efficient for discrimination. The projection is found by solving the generalized eigenvalue problem

$$S_B \mathbf{w} = \lambda S_W \mathbf{w}$$

where  $S_B$  and  $S_W$  are defined in section 2. Suppose there are  $l$  classes. The  $i^{\text{th}}$  class contains  $n_i$  sample points. Let  $\mathbf{m}^{(i)}$  denote the average vector of the  $i^{\text{th}}$  class. Let  $\mathbf{x}^{(i)}$  denote the random vector associated to the  $i^{\text{th}}$  class and  $\mathbf{x}_j^{(i)}$  denote the  $j^{\text{th}}$  sample point in the  $i^{\text{th}}$  class. We can rewrite the matrix  $S_W$  as follows:

$$\begin{aligned} S_W &= \sum_{i=1}^l \left( \sum_{j=1}^{n_i} (\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)}) (\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})^T \right) \\ &= \sum_{i=1}^l \left( \sum_{j=1}^{n_i} (\mathbf{x}_j^{(i)} (\mathbf{x}_j^{(i)})^T - \mathbf{m}^{(i)} (\mathbf{x}_j^{(i)})^T - \mathbf{x}_j^{(i)} (\mathbf{m}^{(i)})^T + \mathbf{m}^{(i)} (\mathbf{m}^{(i)})^T) \right) \\ &= \sum_{i=1}^l \left( \sum_{j=1}^{n_i} \mathbf{x}_j^{(i)} (\mathbf{x}_j^{(i)})^T - n_i \mathbf{m}^{(i)} (\mathbf{m}^{(i)})^T \right) \\ &= \sum_{i=1}^l \left( X_i X_i^T - \frac{1}{n_i} (\mathbf{x}_1^{(i)} + \dots + \mathbf{x}_{n_i}^{(i)}) (\mathbf{x}_1^{(i)} + \dots + \mathbf{x}_{n_i}^{(i)})^T \right) \\ &= \sum_{i=1}^l \left( X_i X_i^T - \frac{1}{n_i} X_i (\mathbf{e}_i \mathbf{e}_i^T) X_i^T \right) \\ &= \sum_{i=1}^l X_i L_i X_i^T \end{aligned}$$

where  $X_i L_i X_i^T$  is the data covariance matrix of the  $i^{\text{th}}$  class, and  $X_i = [\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \dots, \mathbf{x}_{n_i}^{(i)}]$  is a  $d \times n_i$  matrix.  $L_i = I - 1/n_i \mathbf{e}_i \mathbf{e}_i^T$  is a  $n_i \times n_i$  matrix where  $I$  is the identity matrix and  $\mathbf{e}_i = (1, 1, \dots, 1)^T$  is an  $n_i$  dimensional vector. To further simplify the above equation, we define:

$$X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$$

$$W_{ij} = \begin{cases} 1/n_k & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ both belong to the } k^{\text{th}} \text{ class} \\ 0 & \text{otherwise} \end{cases}$$

$$L = I - W$$

Thus, we get:

$$S_W = XLX^T$$

It is interesting to note that we could regard the matrix  $W$  as the weight matrix of a graph with data points as its nodes. Specifically,  $W_{ij}$  is the weight of the edge  $(\mathbf{x}_i, \mathbf{x}_j)$ .  $W$  reflects the class relationships of the data points. The matrix  $L$  is thus called *graph Laplacian*, which plays key role in LPP [9].

Similarly, we can compute the matrix  $S_B$  as follows:

$$\begin{aligned} S_B &= \sum_{i=1}^l n_i (\mathbf{m}^{(i)} - \mathbf{m})(\mathbf{m}^{(i)} - \mathbf{m})^T \\ &= \left( \sum_{i=1}^l n_i \mathbf{m}^{(i)} (\mathbf{m}^{(i)})^T \right) - 2\mathbf{m} \left( \sum_{i=1}^l n_i \mathbf{m}^{(i)} \right) + \left( \sum_{i=1}^l n_i \right) \mathbf{m} \mathbf{m}^T \\ &= \left( \sum_{i=1}^l \frac{1}{n_i} (\mathbf{x}_1^{(i)} + \dots + \mathbf{x}_{n_i}^{(i)}) (\mathbf{x}_1^{(i)} + \dots + \mathbf{x}_{n_i}^{(i)})^T \right) - 2n\mathbf{m} \mathbf{m}^T + n\mathbf{m} \mathbf{m}^T \\ &= \left( \sum_{i=1}^l \sum_{j,k=1}^{n_i} \frac{1}{n_i} \mathbf{x}_j^{(i)} (\mathbf{x}_k^{(i)})^T \right) - 2n\mathbf{m} \mathbf{m}^T + n\mathbf{m} \mathbf{m}^T \\ &= XWX^T - 2n\mathbf{m} \mathbf{m}^T + n\mathbf{m} \mathbf{m}^T \\ &= XWX^T - n\mathbf{m} \mathbf{m}^T \\ &= XWX^T - X \left( \frac{1}{n} \mathbf{e} \mathbf{e}^T \right) X^T \\ &= X \left( W - \frac{1}{n} \mathbf{e} \mathbf{e}^T \right) X^T \\ &= X \left( W - I + I - \frac{1}{n} \mathbf{e} \mathbf{e}^T \right) X^T \\ &= -XLX^T + X \left( I - \frac{1}{n} \mathbf{e} \mathbf{e}^T \right) X^T \\ &= -XLX^T + C \end{aligned}$$

where  $\mathbf{e} = (1, 1, \dots, 1)^T$  is a  $n$  dimensional vector and  $C = X \left( I - \frac{1}{n} \mathbf{e} \mathbf{e}^T \right) X^T$  is the data covariance matrix.

Thus, the generalized eigenvector problem of LDA can be written as follows:

$$\begin{aligned}
S_B \mathbf{w} &= \lambda S_W \mathbf{w} \\
\Rightarrow (C - XLX^T) \mathbf{w} &= \lambda XLX^T \mathbf{w} \\
\Rightarrow C \mathbf{w} &= (1 + \lambda) XLX^T \mathbf{w} \\
\Rightarrow XLX^T \mathbf{w} &= \frac{1}{1 + \lambda} C \mathbf{w}
\end{aligned}$$

Thus, the projections of LDA can be obtained by solving the following generalized eigenvalue problem,

$$XLX^T \mathbf{w} = \lambda C \mathbf{w}$$

The optimal projections correspond to the eigenvectors associated with the smallest eigenvalues. If the sample mean of the data set is zero, the covariance matrix is simply  $XX^T$  which is close to the matrix  $XD X^T$  in the LPP algorithm. Our analysis shows that LDA actually aims to preserve discriminating information and global geometrical structure. Moreover, LDA has a similar form to LPP. However, LDA is supervised while LPP can be performed in either supervised or unsupervised manner.

**Proposition 1: The rank of  $L$  is  $n - c$ .**

Proof: Without loss of generality, we assume that the data points are ordered according to which class they are in, so that  $\{\mathbf{x}_1, \dots, \mathbf{x}_{n_1}\}$  are in the first class,  $\{\mathbf{x}_{n_1+1}, \dots, \mathbf{x}_{n_1+n_2}\}$  are in the second class, etc.

Thus, we can write  $L$  as follows:

$$L = \begin{bmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_c \end{bmatrix}$$

where  $L_i$  is a square matrix,

$$L_i = \begin{bmatrix} 1 - \frac{1}{n_i} & -\frac{1}{n_i} & \dots & -\frac{1}{n_i} \\ -\frac{1}{n_i} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -\frac{1}{n_i} \\ -\frac{1}{n_i} & \dots & -\frac{1}{n_i} & 1 - \frac{1}{n_i} \end{bmatrix}$$

By adding all but the first column vectors to the first column vector and then subtracting the first row vector from any other row vectors, we get the following matrix

$$\begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix}$$

whose rank is  $n_i - 1$ . Therefore, the rank of  $L_i$  is  $n_i - 1$  and hence the rank of  $L$  is  $n - c$ .  $\square$

Proposition 1 tells us that the rank of  $XLX^T$  is at most  $n - c$ . However, in many cases in appearance-based face recognition, the number of pixels in an image (or, the dimensionality of the image space) is larger than  $n - c$ , i.e.  $d > n - c$ . Thus,  $XLX^T$  is singular. In order to overcome the complication of a singular  $XLX^T$ , Belhumeur *et. al.* [1] proposed the Fisherface approach that the face images are projected from the original image space to a subspace with dimensionality  $n - c$  and then LDA is performed in this subspace.

## 6. MANIFOLD WAYS OF VISUAL ANALYSIS

In the above sections, we have described three different linear subspace learning algorithms. The key difference between PCA, LDA and LPP is that, PCA and LDA aim to discover the global structure of the Euclidean space, while LPP aims to discover local structure of the manifold. In this Section, we discuss the manifold ways of visual analysis.

### 6.1 Manifold Learning via Dimensionality Reduction

In many cases, face images may be visualized as points drawn on a low-dimensional manifold hidden in a high-dimensional ambient space. Specially, we can consider that a sheet of rubber is crumpled into a (high-dimensional) ball. The objective of a dimensionality-reducing mapping is to unfold the sheet and to make its low-dimensional structure explicit. If the sheet is not torn in the process, the mapping is topology-preserving. Moreover, if the rubber is not stretched or compressed, the mapping preserves the metric structure of the original space. In this paper, our objective is to discover the face manifold by a locally topology-preserving mapping for face analysis (representation and recognition).

## 6.2 Learning Laplacianfaces for Representation

LPP is a general method for manifold learning. It is obtained by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the manifold [9]. Therefore, though it is still a linear technique, it seems to recover important aspects of the intrinsic nonlinear manifold structure by preserving local structure. Based on LPP, we describe our Laplacianfaces method for face representation in a locality preserving subspace.

In the face analysis and recognition problem one is confronted with the difficulty that the matrix  $XD X^T$  is sometimes singular. This stems from the fact that sometimes the number of images in the training set ( $n$ ) is much smaller than the number of pixels in each image ( $m$ ). In such a case, the rank of  $XD X^T$  is at most  $n$ , while  $XD X^T$  is an  $m \times m$  matrix, which implies that  $XD X^T$  is singular. To overcome the complication of a singular  $XD X^T$ , we first project the image set to a PCA subspace so that the resulting matrix  $XD X^T$  is nonsingular. Another consideration of using PCA as preprocessing is for noise reduction. This method, we call *Laplacianfaces*, can learn an optimal subspace for face representation and recognition.

The algorithmic procedure of Laplacianfaces is formally stated below:

1. **PCA projection:** We project the image set  $\{\mathbf{x}_i\}$  into the PCA subspace by throwing away the smallest principal components. In our experiments, we kept 98% information in the sense of reconstruction

error. For the sake of simplicity, we still use  $\mathbf{x}$  to denote the images in the PCA subspace in the following steps. We denote by  $W_{PCA}$  the transformation matrix of PCA.

2. **Constructing the nearest-neighbor graph:** Let  $G$  denote a graph with  $n$  nodes. The  $i^{th}$  node corresponds to the face image  $\mathbf{x}_i$ . We put an edge between nodes  $i$  and  $j$  if  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are “close”, i.e.  $\mathbf{x}_i$  is among  $k$  nearest neighbors of  $\mathbf{x}_j$  or  $\mathbf{x}_j$  is among  $k$  nearest neighbors of  $\mathbf{x}_i$ . The constructed nearest-neighbor graph is an approximation of the local manifold structure. Note that, here we do not use the  $\varepsilon$  - neighborhood to construct the graph. This is simply because it is often difficult to choose the optimal  $\varepsilon$  in the real world applications, while  $k$  nearest neighbor graph can be constructed more stably. The disadvantage is that the  $k$  nearest neighbor search will increase the computational complexity of our algorithm. When the computational complexity is a major concern, one can switch to the  $\varepsilon$  - neighborhood.

3. **Choosing the weights:** If node  $i$  and  $j$  are connected, put

$$S_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{t}}$$

where  $t$  is a suitable constant. Otherwise, put  $S_{ij} = 0$ . The weight matrix  $S$  of graph  $G$  models the face manifold structure by preserving local structure. The justification for this choice of weights can be traced back to [3].

4. **Eigenmap:** Compute the eigenvectors and eigenvalues for the generalized eigenvector problem:

$$XLX^T \mathbf{w} = \lambda XD X^T \mathbf{w} \tag{1}$$

where  $D$  is a diagonal matrix whose entries are column (or row, since  $S$  is symmetric) sums of  $S$ ,  $D_{ii} = \sum_j S_{ji}$ .  $L = D - S$  is the Laplacian matrix. The  $i^{th}$  row of matrix  $X$  is  $\mathbf{x}_i$ .

Let  $\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_{k-1}$  be the solutions of equation (1), ordered according to their eigenvalues,  $0 \leq \lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_{k-1}$ . These eigenvalues are equal to or greater than zero, because the matrices  $XLX^T$  and  $XD X^T$  are both symmetric and positive semi-definite. Thus, the embedding is as follows:

$$\mathbf{x} \rightarrow \mathbf{y} = W^T \mathbf{x}$$

$$W = W_{PCA} W_{LPP}$$

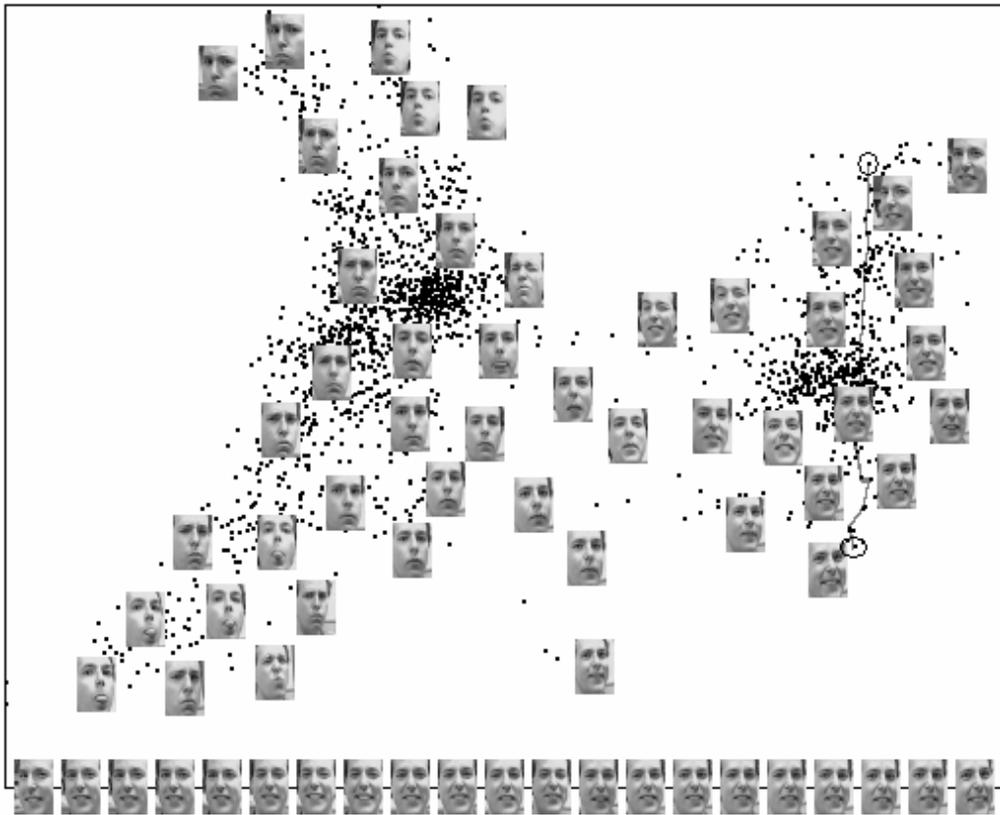
$$W_{LPP} = [\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_{k-1}]$$

where  $\mathbf{y}$  is a  $k$ -dimensional vector.  $W$  is the transformation matrix. This linear mapping best preserves the manifold's estimated intrinsic geometry in a linear sense. The column vectors of  $W$  are the so called *Laplacianfaces*.

### 6.3 Face Manifold Analysis

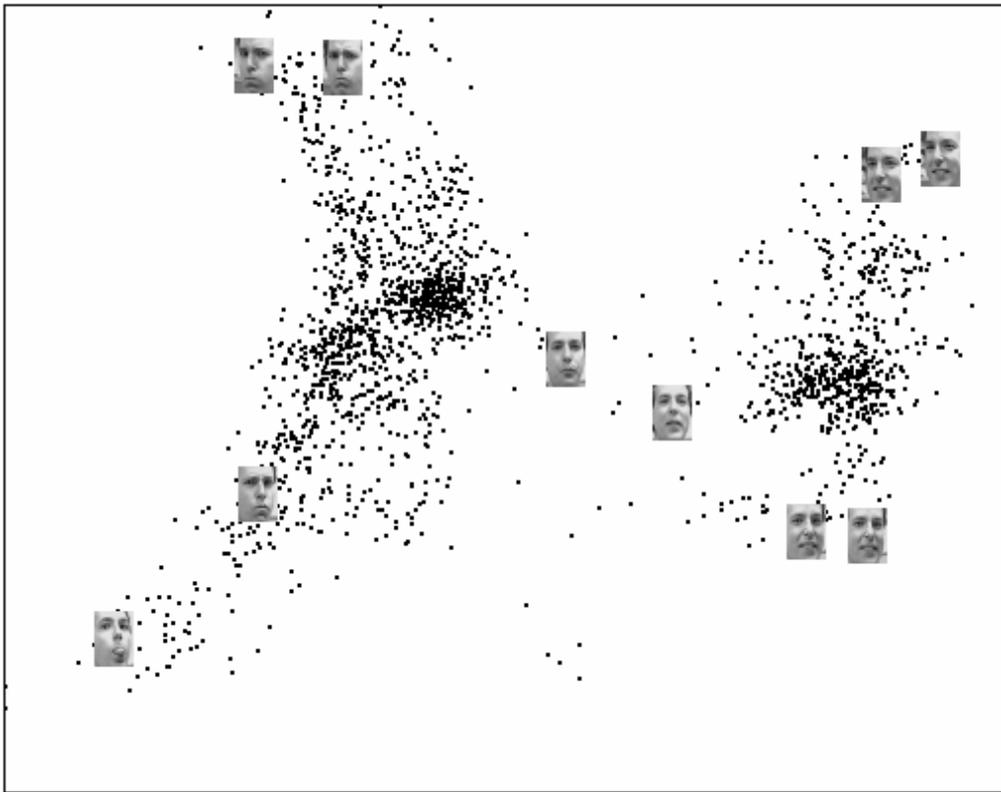
Now consider a simple example of image variability. Imagine that a set of face images are generated while the human face rotates slowly. Intuitively, the set of face images correspond to a continuous curve in image space, since there is only one degree of freedom, viz. the angle of rotation. Thus, we can say that the set of face images are intrinsically one-dimensional. Many recent works [18][19][21][27] have shown that the face images do reside on a low-dimensional submanifold embedded in a high-dimensional ambient space (image space). Therefore, an effective subspace learning algorithm should be able to detect the nonlinear manifold structure. The conventional algorithms, such as PCA and LDA, model the face images in Euclidean space. They effectively see only the Euclidean structure; thus, they fail to detect the intrinsic low-dimensionality.

With its neighborhood preserving character, the *Laplacianfaces* seem to be able to capture the intrinsic face manifold structure to a larger extent. Figure 1 shows an example that the face images with various pose and expression of a person are mapped into two-dimensional subspace. The face image dataset used here is the same as that used in [18]. This dataset contains 1965 face images taken from se-



**Figure 1. Two-dimensional linear embedding of face images by *Laplacianfaces*. As can be seen, the face images are divided into two parts, the faces with open mouth and the faces with closed mouth. Moreover, it can be clearly seen that the pose and expression of human faces change continuously and smoothly, from top to bottom, from left to right. The bottom images correspond to points along the right path (linked by solid line), illustrating one particular mode of variability in pose.**

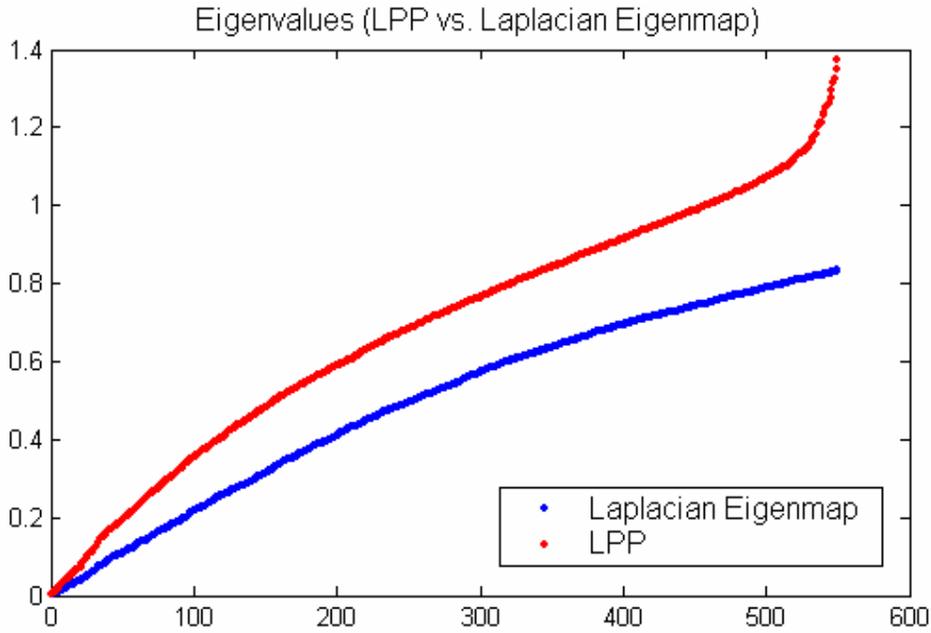
quential frames of a small video. The size of each image is  $20 \times 28$  pixels, with 256 gray levels per pixel. Thus, each face image is represented by a point in the 560-dimensional ambient space. However, these images are believed to come from a submanifold with few degrees of freedom. We leave out 10 samples for testing, and the remaining 1955 samples are used to learn the *Laplacianfaces*. As can be seen, the face images are mapped into a two-dimensional space with continuous change in pose and expression.



**Figure 2. Distribution of the 10 testing samples in the reduced representation subspace. As can be seen, these testing samples optimally find their coordinates which reflect their intrinsic properties, i.e. pose and expression.**

The representative face images are shown in the different parts of the space. The face images are divided into two parts. The left part includes the face images with open mouth, and the right part includes the face images with closed mouth. This is because in trying to preserve local structure in the embedding, the *Laplacianfaces* implicitly emphasizes the natural clusters in the data. Specifically, it makes the neighboring points in the image face nearer in the face space, and faraway points in the image face farther in the face space. The bottom images correspond to points along the right path (linked by solid line), illustrating one particular mode of variability in pose.

The ten testing samples can be simply located in the reduced representation space by the *Laplacianfaces* (column vectors of the matrix  $W$ ). Figure 2 shows the result. As can be seen, these testing



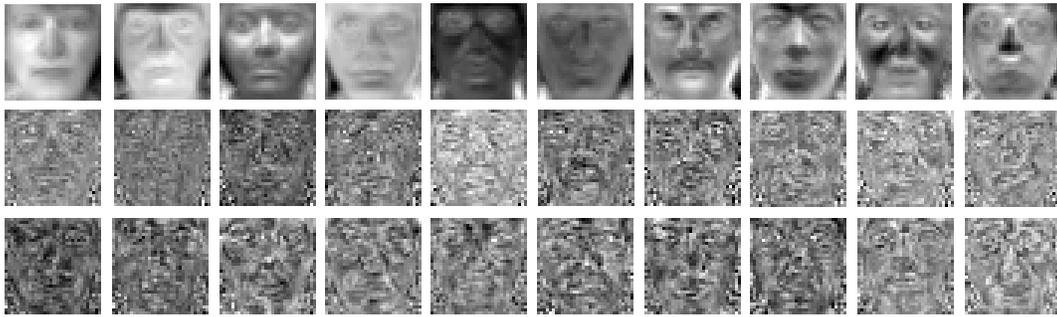
**Figure 3. The eigenvalues of LPP and Laplacian Eigenmap.**

samples optimally find their coordinates which reflect their intrinsic properties, i.e. pose and expression. This observation tells us that the *Laplacianfaces* are capable of capturing the intrinsic face manifold structure to some extent.

Recall that both Laplacian Eigenmap and LPP aim to find a map which preserves the local structure. Their objective function is as follows:

$$\min_f \sum_{ij} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 S_{ij}$$

The only difference between them is that, LPP is linear while Laplacian Eigenmap is non-linear. Since they have the same objective function, it would be important to see to what extent LPP can approximate Laplacian Eigenmap. This can be evaluated by comparing their eigenvalues. Figure 3 shows the eigenvalues computed by the two methods. As can be seen, the eigenvalues of LPP is consistently greater than those of Laplacian Eigenmaps, but the difference between them is small. The difference gives a measure of the degree of the approximation.



**Figure 4. The first 10 Eigenfaces (first row), Fisherfaces (second row) and Laplacianfaces (third row) calculated from the face images in the YALE database.**

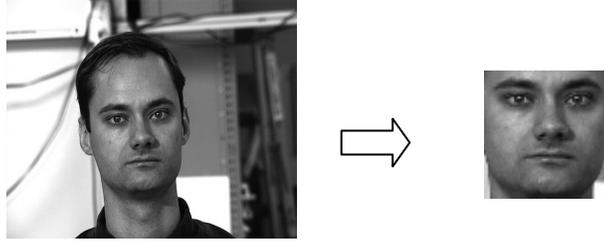
## 7. EXPERIMENTAL RESULTS

Some simple synthetic examples given in [9] show that LPP can have more discriminating power than PCA and be less sensitive to outliers. In this section, several experiments are carried out to show the effectiveness of our proposed Laplacianfaces method for face representation and recognition.

### 7.1 Face Representation Using Laplacianfaces

As we described previously, a face image can be represented as a point in image space. A typical image of size  $m \times n$  describes a point in  $m \times n$ -dimensional image space. However, due to the unwanted variations resulting from changes in lighting, facial expression, and pose, the image space might not be an optimal space for visual representation.

In Section 3, we have discussed how to learn a locality preserving face subspace which is insensitive to outlier and noise. The images of faces in the training set are used to learn such a locality preserving subspace. The subspace is spanned by a set of eigenvectors of equation (1), i.e.  $\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_{k-1}$ . We can display the eigenvectors as images. These images may be called *Laplacianfaces*. Using the Yale face database as the training set, we present the first 10 Laplacianfaces in Figure 4, together with Eigenfaces and Fisherfaces. A face image can be mapped into the locality preserving subspace by using the Laplacianfaces. It is interesting to note that the Laplacianfaces are somehow similar to Fisherfaces.



**Figure 5. The original face image and the cropped image.**

## 7.2 Face Recognition Using Laplacianfaces

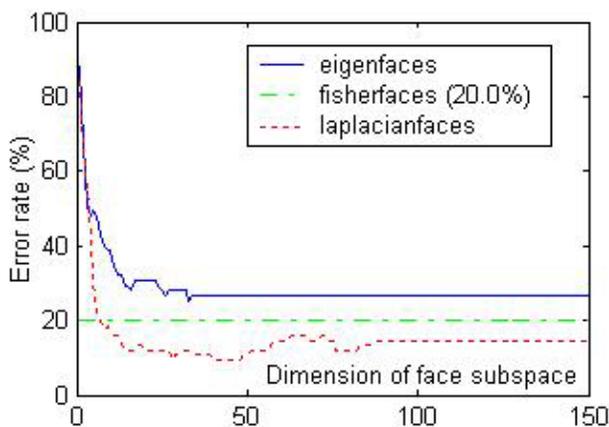
Once the Laplacianfaces are created, face recognition [2][14][28][29] becomes a pattern classification task. In this section, we investigate the performance of our proposed Laplacianfaces method for face recognition. The system performance is compared with the Eigenfaces method [28] and the Fisherfaces method [2], two of the most popular linear methods in face recognition.

In this study, three face databases were tested. The first one is the PIE (pose, illumination, and expression) database from CMU [25], the second one is the Yale database [30], and the third one is the MSRA database collected at the Microsoft Research Asia. In all the experiments, preprocessing to locate the faces was applied. Original images were normalized (in scale and orientation) such that the two eyes were aligned at the same position. Then, the facial areas were cropped into the final images for matching. The size of each cropped image in all the experiments is  $32 \times 32$  pixels, with 256 grey levels per pixel. Thus, each image is represented by a 1024-dimensional vector in image space. No further preprocessing is done. Figure 5 shows an example of the original face image and the cropped image. Different pattern classifiers have been applied for face recognition, including nearest-neighbor [2], Bayesian [15], Support Vector Machine [17], etc. In this paper, we apply the nearest-neighbor classifier for its simplicity.

In short, the recognition process has three steps. First, we calculate the Laplacianfaces from the training set of face images; then the new face image to be identified is projected into the face subspace spanned by the Laplacianfaces; finally, the new face image is identified by a nearest-neighbor classifier.

**Table 1. Performance comparison on the Yale database**

Approach	Dims	Error Rate
Eigenfaces	33	25.3%
Fisherfaces	14	20.0%
<b>Laplacianfaces</b>	<b>28</b>	<b>11.3%</b>



**Figure 6. Recognition accuracy vs. dimensionality reduction on Yale database.**

### 7.2.1 YALE DATABASE

The Yale face database [30] was constructed at the Yale Center for Computational Vision and Control. It contains 165 grayscale images of 15 individuals. The images demonstrate variations in lighting condition (left-light, center-light, right-light), facial expression (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses.

A random subset with six images per individual (hence 90 images in total) was taken with labels to form the training set. The rest of the database was considered to be the testing set. The training samples were used to learn the Laplacianfaces. The testing samples were then projected into the low-dimensional representation subspace. Recognition was performed using a nearest neighbor classifier. We averaged the results over 20 random splits.



**Figure 7. The sample cropped face images of one individual from PIE database. The original face images are taken under varying pose, illumination, and expression.**

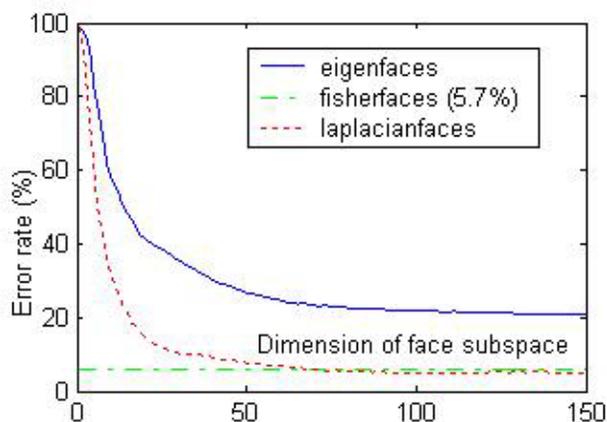
**Table 2. Performance comparison on the PIE database**

Approach	Dims	Error Rate
Eigenfaces	150	20.6%
Fisherfaces	67	5.7%
<b>Laplacianfaces</b>	<b>110</b>	<b>4.6%</b>

Note that, for LDA, there are at most  $c-1$  nonzero generalized eigenvalues, and so an upper bound on the dimension of the reduced space is  $c-1$ , where  $c$  is the number of individuals [2]. In general, the performance of the Eigenfaces method and the Laplacianfaces method varies with the number of dimensions. We show the best results obtained by Fisherfaces, Eigenfaces, and Laplacianfaces.

The recognition results are shown in Table 1. It is found that the Laplacianfaces method significantly outperforms both Eigenfaces and Fisherfaces methods. The recognition approaches the best results with 14, 28, 33 dimensions for Fisherfaces, Laplacianfaces, and Eigenfaces respectively. The error rates of Fisherfaces, Laplacianfaces, and eigenfaces are 20%, 11.3%, and 25.3% respectively. The corresponding face subspaces are called optimal face subspace for each method. There is no significant improvement if more dimensions are used. Figure 6 shows the plots of error rate vs. dimensionality reduction.

### 7.2.2 PIE DATABASE



**Figure 8. Recognition accuracy vs. dimensionality reduction on PIE database**



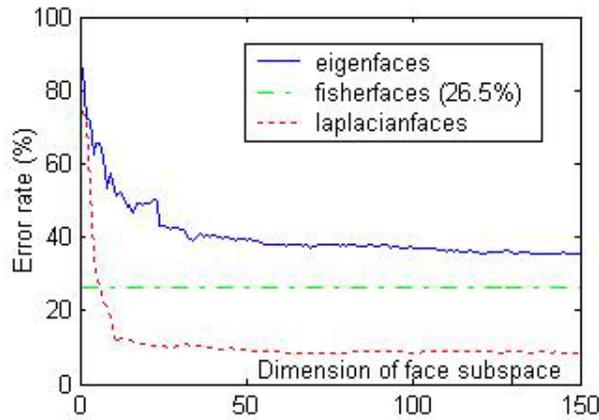
**Figure 9. The sample cropped face images of 8 individuals from MSRA database. The face images in the first row are taken in the first session, which are used for training. The face images in the second row are taken in the second session, which are used for testing. The two images in the same column are corresponding to the same individual.**

The CMU PIE face database contains 68 subjects with 41,368 face images as a whole. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination and expression. We used 170 face images for each individual in our experiment, 85 for training and the other 85 for testing. We averaged the results over 20 random splits. Figure 7 shows some of the faces with pose, illumination and expression variations in the PIE database.

Table 2 shows the recognition results. As can be seen, Fisherfaces performs comparably to our algorithm on this database, while Eigenfaces performs poorly. The error rate for Laplacianfaces, Fisher-

**Table 3. Performance comparison on the MSRA database**

Approach	Dims	Error Rate
Eigenfaces	142	35.4%
Fisherfaces	11	26.5%
<b>Laplacianfaces</b>	<b>66</b>	<b>8.2%</b>

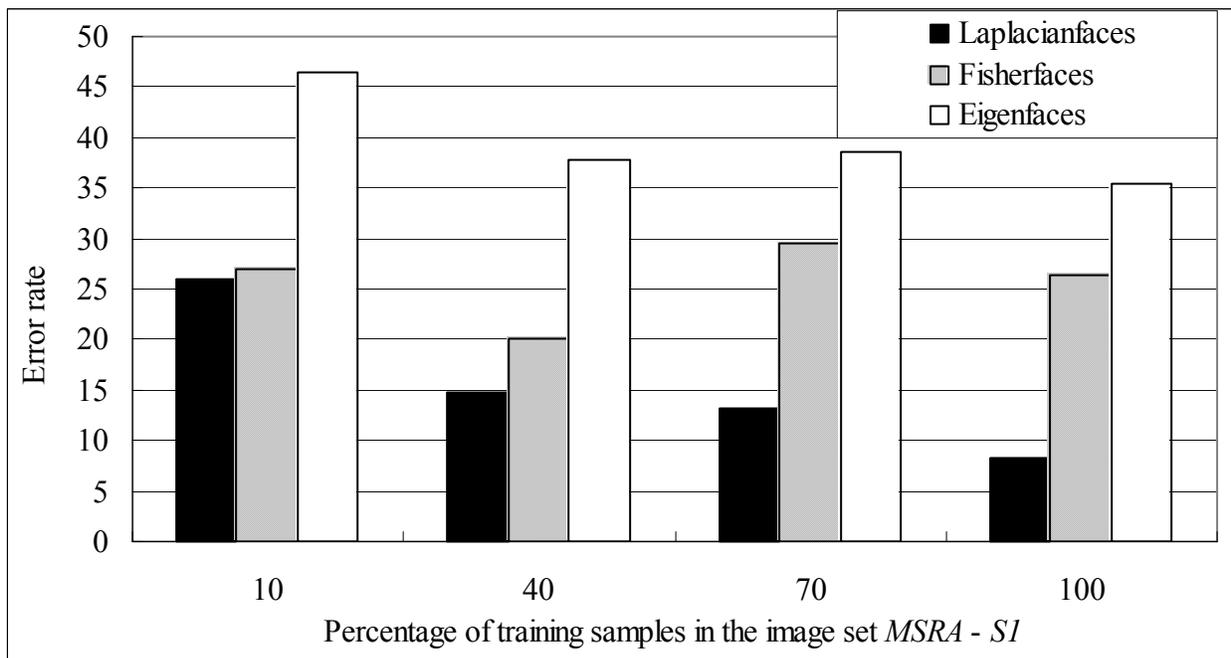


**Figure 10. Recognition accuracy vs. dimensionality reduction on MSRA database**

faces, and Eigenfaces are 4.6%, 5.7%, and 20.6%, respectively. Figure 8 shows a plot of error rate vs. dimensionality reduction. As can be seen, the error rate of our Laplacianfaces method decreases fast as the dimensionality of the face subspace increases, and achieves the best result with 110 dimensions. There is no significant improvement if more dimensions are used. Eigenfaces achieves the best result with 150 dimensions. For Fisherfaces, the dimension of the face subspace is bounded by  $c-1$ , and it achieves the best result with  $c-1$  dimensions. The dashed horizontal line in the figure shows the best result obtained by Fisherfaces.

### 7.2.3 MSRA DATABASE

This database was collected at Microsoft Research Asia. It contains 12 individuals, captured in two different sessions with different backgrounds and illuminations. 64 to 80 face images were collected



**Figure 11. Performance comparison on the MSRA database with different number of training samples. The training samples are from the set *MSRA-S1*. The images in *MSRA-S2* are used for testing exclusively. As can be seen, our Laplacianfaces method takes advantage of more training samples. The error rate reduces from 26.04% with 10% training samples to 8.17% with 100% training samples. There is no evidence for fisherfaces that it can take advantage of more training samples.**

for each individual in each session. All the faces are frontal. Figure 9 shows the sample cropped face images from this database. In this test, one session was used for training and the other was used for testing. Table 3 shows the recognition results. Laplacianfaces method has lower error rate (8.2%) than those of eigenfaces (35.4%) and fisherfaces (26.5%). Figure 10 shows a plot of error rate vs. dimensionality reduction.

#### 7.2.4 IMPROVEMENT WITH THE NUMBER OF TRAINING SAMPLES

In the above subsections, we have shown that our Laplacianfaces approach outperforms the Eigenfaces and Fisherfaces approaches on three different face databases. In this section, we evaluate the per-

**Table 4. Recognition error rate with different number of training samples**

	Percentage of training samples in <i>MSRA-S1</i>			
	10%	40%	70%	100%
<b>Laplacianfaces</b>	<b>26.04%</b>	<b>14.71%</b>	<b>13.15%</b>	<b>8.17%</b>
Fisherfaces	26.96%	20.05%	29.43%	26.43%
Eigenfaces	46.48%	37.76%	38.54%	35.42%

formances of these three approaches with different number of training samples. We expect that a good algorithm should be able to take advantage of more training samples.

We used the MSRA database for this experiment. Let *MSRA-S1* denote the image set taken in the first session, and *MSRA-S2* denote the image set taken in the second session. *MSRA-S2* was exclusively used for testing. 10%, 40%, 70%, and 100% face images were randomly selected from *MSRA-S1* for training. For each of the first three cases, we repeated 20 times and computed the error rate on average. In Figure 11 and Table 4 we show the experimental results of the improvement of the performance of our algorithm with the number of training samples. As can be seen, our algorithm takes advantage of more training samples. The performance improves significantly as the number of training sample increases. The error rate of recognition reduces from 26.04% with 10% training samples to 8.17% with 100% training samples. This is due to the fact that, our algorithm aims at discovering the underlying face manifold embedded in the ambient space and more training samples help to recover the manifold structure. Eigenfaces also takes advantage of more training samples to some extent. The error rate of Eigenfaces reduces from 46.48% with 10% training samples to 35.42% with 100% training samples. But its performance starts to converge at 40% (training samples) and little improvement is obtained if more training samples are used. For Fisherfaces, there is no convincing evidence that it can take advantage of more training samples.

### 7.3 Discussions

Three experiments on three databases have been systematically performed. These experiments reveal a number of interesting points:

1. All these three approaches performed better in the optimal face subspace than in the original image space.
2. In all the three experiments, Laplacianfaces consistently performs better than Eigenfaces and Fisherfaces. Especially, it significantly outperforms Fisherfaces and Eigenfaces on Yale database and MSRA database. These experiments also show that our algorithm is especially suitable for frontal face images. Moreover, our algorithm takes advantage of more training samples, which is important to the real world face recognition systems.
3. Comparing to the Eigenfaces method, the Laplacianfaces method encodes more discriminating information in the low-dimensional face subspace by preserving local structure which is more important than the global structure for classification, especially when nearest neighbor like classifiers are used. In fact, if there is a reason to believe that Euclidean distances ( $\|\mathbf{x}_i - \mathbf{x}_j\|$ ) are meaningful only if they are small (local), then our algorithm finds a projection that respects such a belief. Another reason is that, as we show in Figure 1, the face images probably reside on a nonlinear manifold. Therefore, an efficient and effective subspace representation of face images should be capable of charactering the nonlinear manifold structure, while the Laplacianfaces are exactly derived by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the face manifold. By discovering the face manifold structure, Laplacianfaces can identify the person with different expressions, poses, and lighting conditions.

### 8. CONCLUSION AND FUTURE WORK

The manifold ways of face analysis (representation and recognition) is introduced in this paper in order to detect the underlying nonlinear manifold structure in the manner of linear subspace learning. To

the best of our knowledge, this is the first devoted work on face representation and recognition which explicitly considers the manifold structure. The manifold structure is approximated by the adjacency graph computed from the data points. Using the notion of the Laplacian of the graph, we then compute a transformation matrix which maps the face images into a face subspace. We call this the Laplacianfaces approach. The Laplacianfaces are obtained by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the face manifold. This linear transformation optimally preserves local manifold structure. Theoretical analysis of the LPP algorithm and its connections to PCA and LDA are provided. Experimental results on PIE, Yale and MSRA databases show the effectiveness of our method.

One of the central problems in face manifold learning is to estimate the intrinsic dimensionality of the *nonlinear* face manifold, or, degrees of freedom. We know that the dimensionality of the manifold is equal to the dimensionality of the local tangent space. Some previous works [33][34] show that the local tangent space can be approximated using points in a neighbor set. Therefore, one possibility is to estimate the dimensionality of the tangent space.

Another possible extension of our work is to consider the use of the unlabeled samples. It is important to note that the work presented here is a general method for face analysis (face representation and recognition) by discovering the underlying face manifold structure. Learning the face manifold (or, learning Laplacianfaces) is essentially an unsupervised learning process. And in many practical cases, one finds a wealth of easily available unlabeled samples. These samples might help to discover the face manifold. For example, in [4], it is shown how unlabeled samples are used for discovering the manifold structure and hence improving the classification accuracy. Since the face images are believed to reside on a sub-manifold embedded in a high-dimensional ambient space, we believe that the unlabeled samples are of great value. We are currently exploring these problems in theory and practice.

## REFERENCES

- [1] A. U. Batur and M. H. Hayes, "Linear Subspace for Illumination Robust Face Recognition", *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Hawaii, Dec. 11-13, 2001.
- [2] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, No. 7, 1997, pp. 711-720.
- [3] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering", *Advances in Neural Information Processing System 15*, Vancouver, British Columbia, Canada, 2001.
- [4] M. Belkin and P. Niyogi, "Using Manifold Structure for Partially Labeled Classification", *Advances in Neural Information Processing System 15*, Vancouver, British Columbia, Canada, 2002.
- [5] M. Brand, "Charting a Manifold", *Advances in Neural Information Processing Systems*, Vancouver, Canada, 2002.
- [6] Fan R. K. Chung, Spectral Graph Theory, *Regional Conferences Series in Mathematics*, number 92, 1997.
- [7] Y. Chang, C. Hu and M. Turk, "Manifold of Facial Expression", *Proc. IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, Nice, France, Oct 2003.
- [8] R. Gross, J. Shi, and J. Cohn, "Where to go with Face Recognition", *Third Workshop on Empirical Evaluation Methods in Computer Vision*, Kauai, Hawaii, December 11-13, 2001.
- [9] Xiaofei He and Partha Niyogi, "Locality Preserving Projections", *Advances in Neural Information Processing Systems*, Vancouver, Canada, 2003.
- [10] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Video-Based Face Recognition Using Probabilistic Appearance Manifolds", *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 313-320, vol. 1, 2003.

- [11] A. Levin and A. Shashua, "Principal Component Analysis Over Continuous Subspaces and Intersection of Half-Spaces", *Proc. of the European Conference on Computer Vision (ECCV)*, Copenhagen, Denmark, May 2002.
- [12] S. Z. Li, X. W. Hou, H. J. Zhang, Q. S. Cheng, "Learning Spatially Localized, Parts-Based Representation", *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Hawaii, Dec. 11-13, 2001.
- [13] Q. Liu, R. Huang, H. Lu, and S. Ma, "Face Recognition Using Kernel Based Fisher Discriminant Analysis", in *Proc. of the fifth Int. Conf. on Automatic Face and Gesture Recognition*, Washington D. C., May 2002.
- [14] A. M. Martinez and A. C. Kak, "PCA versus LDA", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(2):228-233, 2001.
- [15] B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Representation", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol 19, pp. 696-710, 1997.
- [16] H. Murase and S. K. Nayar, "Visual Learning and Recognition of 3-D objects from appearance", *International Journal of Computer Vision*, 14:5-24, 1995.
- [17] P. J. Phillips, "Support Vector Machines Applied to Face Recognition". In *Advances in Neural Information Processing Systems* 11, pp. 803-809, 1998.
- [18] Sam T. Roweis, and Lawrence K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding", *Science*, vol 290, 22 December 2000.
- [19] Sam Roweis, Lawrence Saul and Geoff Hinton, "Global Coordination of Local Linear Models", *Advances in Neural Information Processing System* 14, 2001.
- [20] Lawrence K. Saul and Sam T. Roweis, "Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds", *Journal of Machine Learning Research*, v4, pp. 119-155, 2003.
- [21] H. Sebastian Seung and Daniel D. Lee, "The Manifold Ways of Perception", *Science*, vol 290, 22 December 2000.

- [22] T. Shakunaga, and K. Shigenari, "Decomposed Eigenface for Face Recognition under Various Lighting Conditions", *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Hawaii, Dec. 11-13, 2001.
- [23] A. Shashua, A. Levin and S. Avidan, "Manifold Pursuit: A New Approach to Appearance Based Recognition", *International Conference on Pattern Recognition (ICPR)*, August 2002, Quebec, Canada.
- [24] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22 (2000), 888-905.
- [25] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression (PIE) Database", in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, May, 2002.
- [26] L. Sirovich and M. Kirby, "Low-Dimensional Procedure for the Characterization of Human Faces", *Journal of the Optical Society of America A*, vol. 4, pp. 519-524, 1987.
- [27] J. B. Tenenbaum, Vin de Silva, and Johh C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction", *Science*, vol 290, 22 December 2000.
- [28] M. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces", *IEEE Conference on Computer Vision and Pattern Recognition*, Maui, Hawaii, 1991.
- [29] L. Wiskott, J.M. Fellous, N. Kruger, and C.v.d. Malsburg, "Face Recognition by Elastic Bunch Graph Matching", *IEEE Trans. PAMI*, 19:775-779, 1997.
- [30] Yale Univ. Face Database, <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>, 2002.
- [31] Ming-Hsuan Yang, "Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods", in *Proc. of the fifth Int. Conf. on Automatic Face and Gesture Recognition*, Washington D. C., May 2002.

- [32]J. Yang, Y. Yu, and W. Kunz, “An Efficient LDA Algorithm for Face Recognition”, *The sixth International Conference on Control, Automation, Robotics and Vision*, Singapore, 2000.
- [33]Hongyuan Zha and Zhenyue Zhang, “Isometric Embedding and Continuum ISOMAP”, in *Proc. Of the Twentieth International Conference on Machine Learning*, pp. 864-871, 2003.
- [34]Zhengyue Zhang and Hongyuan Zha, “Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment”, CSE-02-019, Technical Report, CSE, Penn State Univ., 2002.
- [35]W. Zhao, R. Chellappa, and P. J. Phillips, “Subspace Linear Discriminant Analysis for Face Recognition”, *Technical Report CAR-TR-914*, Center for Automation Research, University of Maryland, 1999.